TEST

1. Min and Max in a List in Java

   Solution:

```
import java.util.Arrays;
import java.util.List;
import java.util.Collections;

public class Numbers {
    public static void main(String[] args) {
      List<Integer> numbers = Arrays.asList(3, 8, 2, 6, 4, 9, 18);
      Integer min = Collections.min(numbers);
      Integer max = Collections.max(numbers);

      System.out.println("Minimum: " + min);
      System.out.println("Maximum: " + max);

  }
}
```

   Output:

```
Minimum: 2
Maximum: 18
```

2. Split a List into Two Halves in Java.

   Solution:

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Numbers {
    public static void main(String[] args) {
      List<Integer> numbers = Arrays.asList(1, 4, 5, 9, 10, 11, 15, 8, 21);
```

```
        int midIndex = numbers.size() / 2;
        List<Integer> firstHalf = new ArrayList<>(numbers.subList(0,
    midIndex));
        List<Integer> secondHalf = new
    ArrayList<>(numbers.subList(midIndex, numbers.size()));

        System.out.println("First Half: " + firstHalf);
        System.out.println("Second Half: " + secondHalf);
      }
    }
```

Output:

```
First Half: [1, 4, 5, 9]
Second Half: [10, 11, 15, 8, 21]
```

3. Remove Duplicates from ArrayList in Java.

Solution:

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.LinkedHashSet;

public class Color {
    public static void main(String[] args) {
        List<String> color = Arrays.asList("Pink", "Black", "Yellow", "Pink",
    "Blue");

        LinkedHashSet<String> uniqueNames = new
    LinkedHashSet<>(color);
        List<String> uniqueList = new ArrayList<>(uniqueNames);

        System.out.println("List without duplicates: " + uniqueList);

      }
    }
```

Output:

List without duplicates: [Pink, Black, Yellow, Blue]

4. Add Element at First and Last Position of LinkedList in Java.

Solution:

```
import java.util.LinkedList;

public class Fruits {
    public static void main(String[] args) {
      LinkedList<String> fruits = new LinkedList<>();

      fruits.addFirst("Apple");
      fruits.addLast("Banana");

      System.out.println("LinkedList: " + fruits);
  }
}
```

Output:

LinkedList: [Apple, Banana]